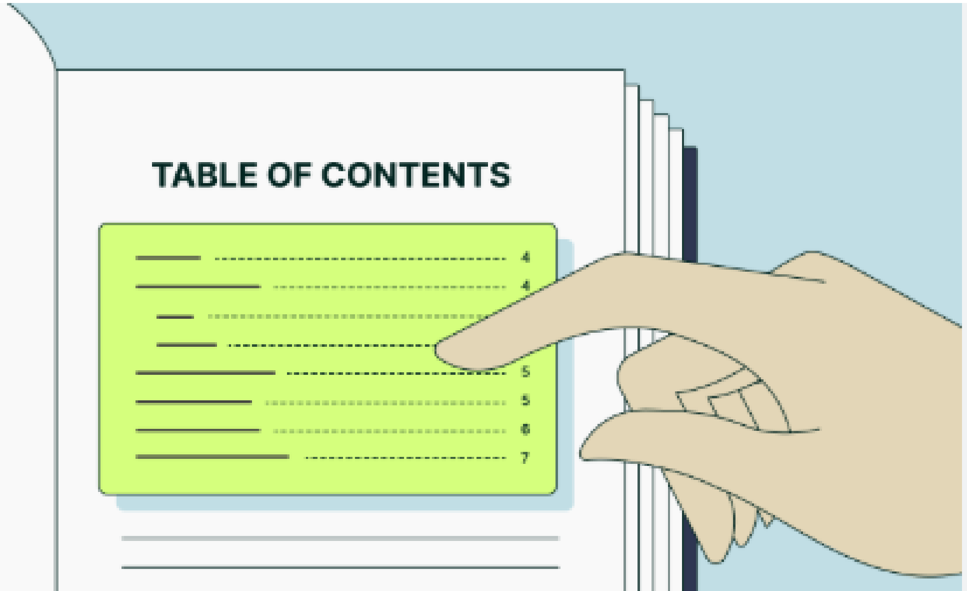
Software Engineering

Unit-1

Introduction to S/w Engineering

Table of Content

- Feasibility Study
- Functional & Non- Functional Requirements
- Requirement Gathering
- Requirement Analysis & Specification
- SRS



Feasibility Study in Software Engineering

Definition

- Analyzes the practicality and viability of a software project.
- Determines whether the project is worth pursuing.

Purpose

- Identifies potential **challenges and risks** early in the process.
- **Ensures** that the software project is ***technically, financially, and operationally feasible.***
- Helps stakeholders ***make informed decisions*** about project initiation.

Contd.

❖ Importance

- ***Identifies*** potential **risks** and provides a clear **roadmap**.
- ***Ensures*** that resources are used efficiently and effectively.
- ***Helps to avoid*** costly project failures.

❖ Types of Feasibility

- **Technical Feasibility**
  - Assesses if the required ***technology and resources exist***.
- **Economic Feasibility**
  - Evaluates ***cost-benefit analysis*** and return on investment.
- **Legal Feasibility**
  - Checks compliance with ***laws and regulations.***
- **Operational Feasibility**
  - Determines if the project meets ***user needs and expectations***
- **Scheduling Feasibility**
  - Assesses if the ***project can be completed on time***.

What is S/w Requirement?

- Software requirement is a ***condition*** *needed by a user to solve a problem or* ***achieve an objective***.
- Requirement should be *clear, correct and well defined.*

There are two types of Requirements:

1. Functional Requirements
2. Non-Functional Requirements

Functional and Non-Functional Requirements

Importance

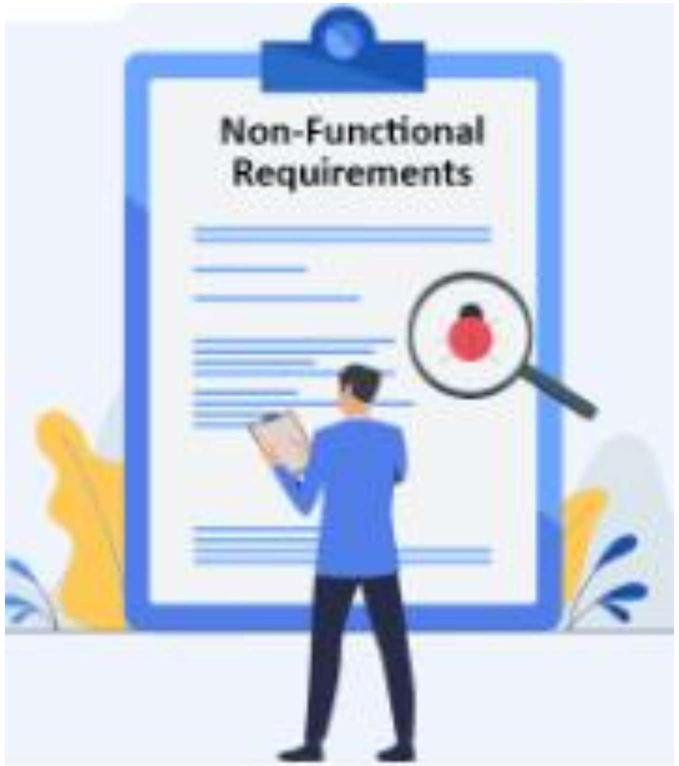
- *Functional* requirements guide ***system capabilities and scope.***
- *Non-functional* requirements ensure ***system meets quality standards***.

□Functional Requirements

- Define the **core features and behaviors of the system**.
- Focus **on what the system should do.**
- **Examples:**
  - User login and authentication.
  - Data processing and calculations.
  - Reporting and data export.
  - Interaction with external systems.

❑ Non-Functional Requirements

- ***Define*** <u>system attributes and quality criteria</u>.
- ***Focus on*** <u>how the system should perform</u>.
- Examples:
  - Performance (e.g. response time, throughput)
  - Security (e.g. encryption, user access controls)
  - Scalability (e.g. ability to handle growing user base)
  - Usability (e.g. user-friendly interface)
  - Reliability (e.g. system fault tolerance)
  - Compliance (e.g. legal or regulatory standards)



Difference

Functional Requirements

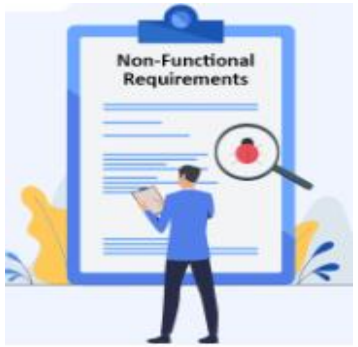
- **Describes** what the system should do i.e. specific functionality or tasks
- **Focuses** on the behaviour and features of the system
- **Defines** the actions and operations of the system

User authentication, data input/output, transaction processing

Non-Functional Requirements

- Describes ***how*** the system should perform i.e. system attribute or quality
- Focuses on the performance, usability, and other quality attributes or constraints
- Defines constraints or conditions under which the system must operate

Scalability security, response time, reliability, maintainability



Functional Requirements Example

define what the system should do and specify specific functionality or behaviour.

Online Banking System

- The system must allow users to log in using a username and password.
- Users must be able to transfer funds between accounts.
- Users should receive an email notification for successful transactions.
- Customers must be able to view their account balance and transaction history.

E-commerce Website

- Users must be able to search for products by name or category.
- The system should allow users to add products to a cart and proceed to checkout.
- The website must send order confirmation emails after a purchase.
- Admins must be able to add, up

Non Functional Requirements Example

define how the system should behave or its operational constraints.

Online Banking System

- The system should handle **5,000 simultaneous users** without degradation in performance.
- Response time for account balance queries must be less than **2 seconds**.
- The system must use **256-bit encryption** for secure transactions.
- The system should have **99.9% uptime** per year.

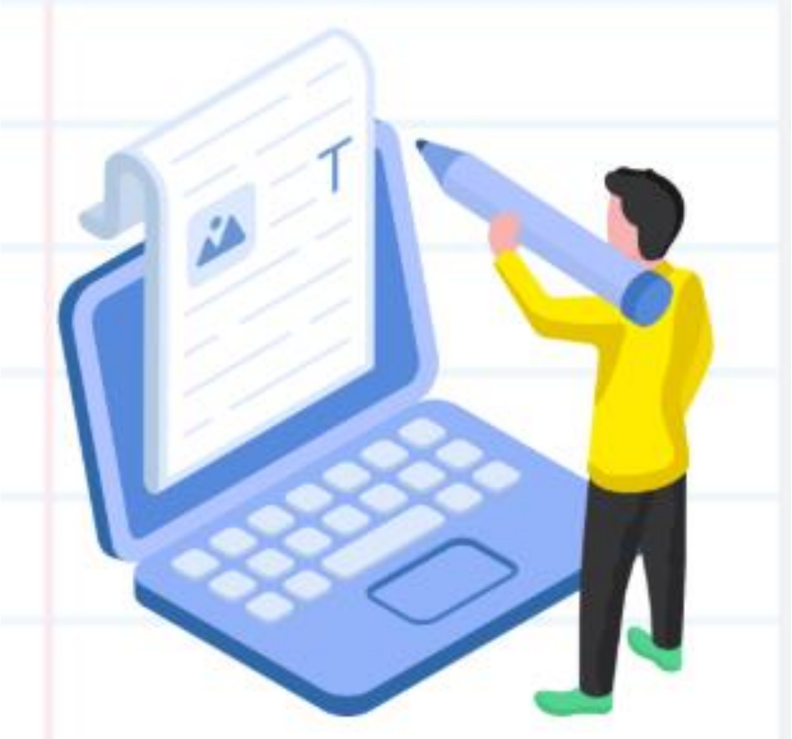
E-commerce Website

- The website must load within **3 seconds** for users on a 4G connection.
- The system must be **scalable** to handle **1 million concurrent users** during flash sales.
- The website should be accessible on desktop and mobile devices (responsive design).
- It must comply with **GDPR (**General Data Protection Regulation) **regulations** for data privacy.

Requirement Gathering

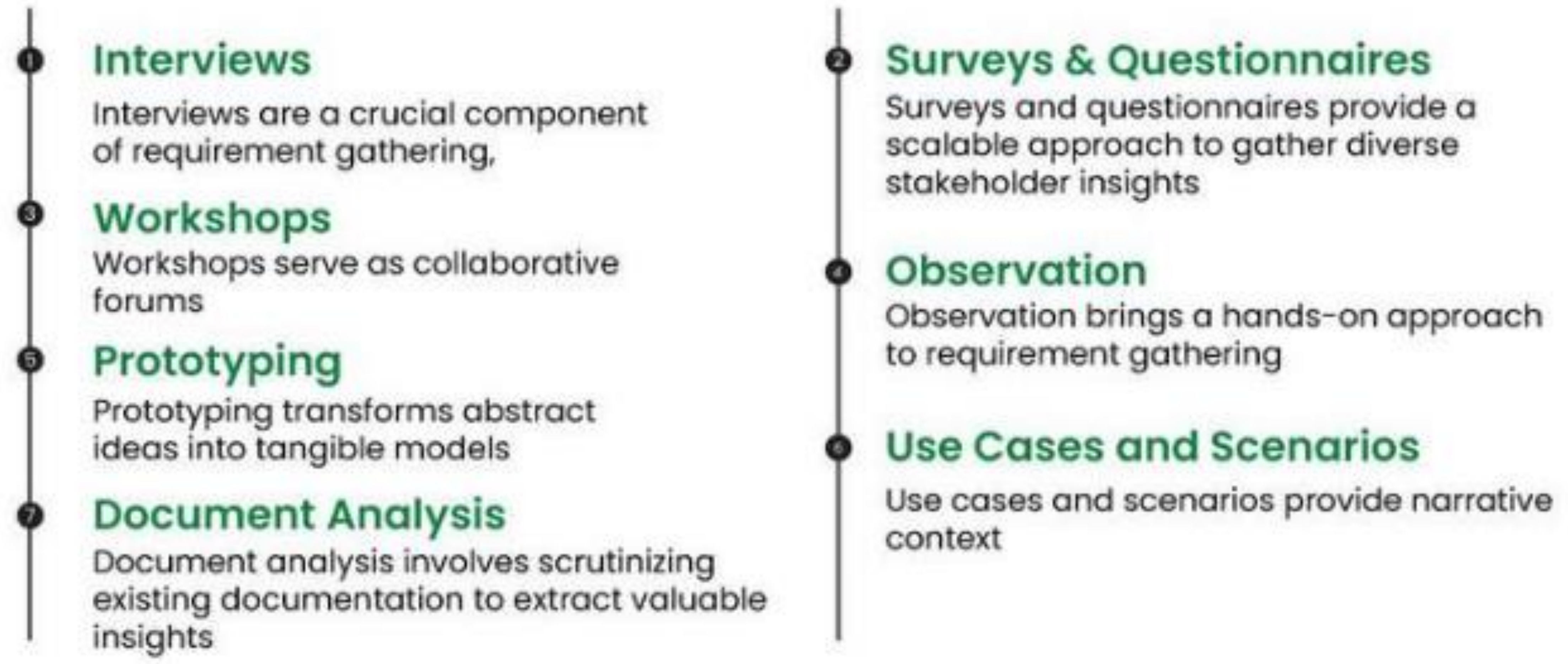
Why Requirement Gathering is Important?

- Clarity of Project Objectives
- Customer Satisfaction
- Scope Definition
- Reduced Misunderstandings
- Risk Mitigation



Contd.

Requirement Gathering Techniques



1 **Interviews**
Interviews are a crucial component of requirement gathering,

3 **Workshops**
Workshops serve as collaborative forums

5 **Prototyping**
Prototyping transforms abstract ideas into tangible models

7 **Document Analysis**
Document analysis involves scrutinizing existing documentation to extract valuable insights

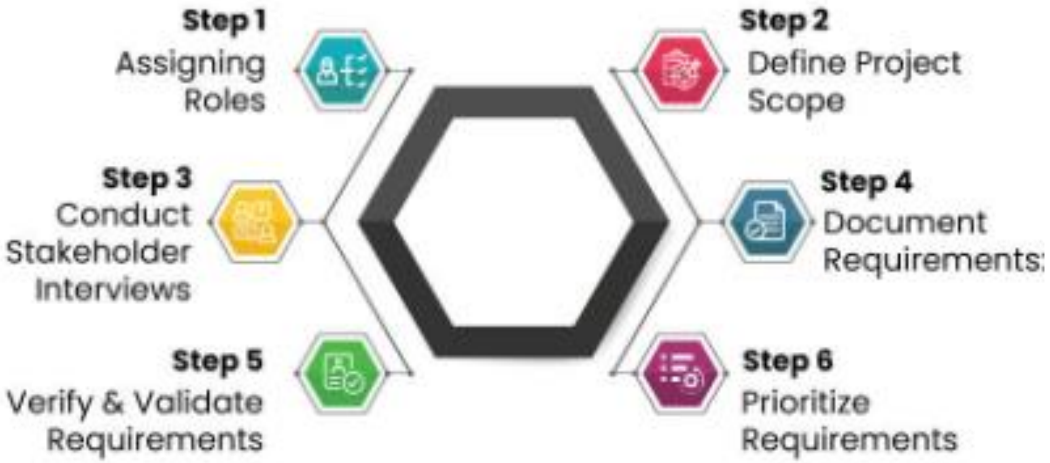
2 **Surveys & Questionnaires**
Surveys and questionnaires provide a scalable approach to gather diverse stakeholder insights

4 **Observation**
Observation brings a hands-on approach to requirement gathering

6 **Use Cases and Scenarios**
Use cases and scenarios provide narrative context

Contd. Requirement Gathering

- Critical phase in the Software Development Life Cycle (SDLC).
- Involves collecting, documenting and managing system requirements.
- Defines features and functionalities of the system or application.
- Ensures project goals align with stakeholder expectations.
- Accurate requirements are key to **project success.**
- Completeness of requirements impacts **overall software quality.**
- Provides a **foundation** for design and development phases.



Contd.

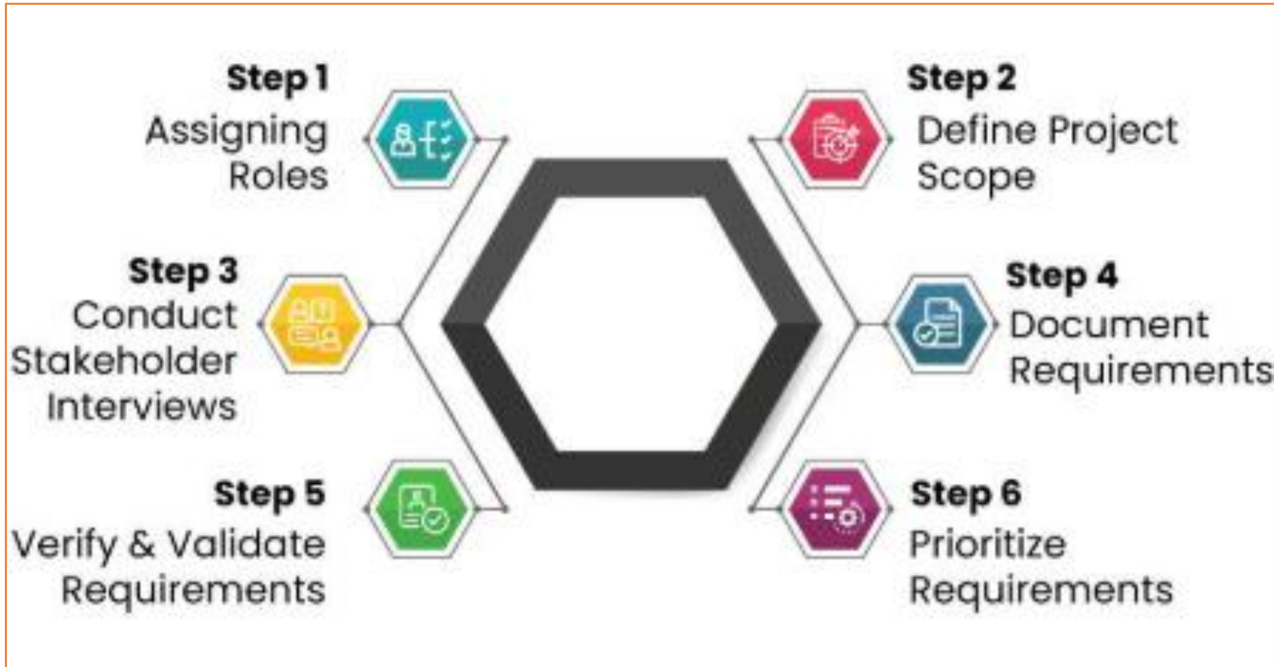
Requirement Gathering Process

Step 1: Assigning Roles

- Identify and engage with all relevant project stakeholders.
- Include end-users, clients, and subject matter experts.
- Understand diverse perspectives for comprehensive requirement gathering.

Step 2: Define Project Scope

- Outline project objectives, boundaries, and limitations clearly.
- Establish a shared understanding of project goals and functionalities.



Contd. Requirement Gathering

Step 3: Conduct Stakeholder Interviews

- Schedule interviews with key stakeholders for insights.
- Use open-ended questions to uncover explicit and implicit needs.

Step 4: Document Requirements

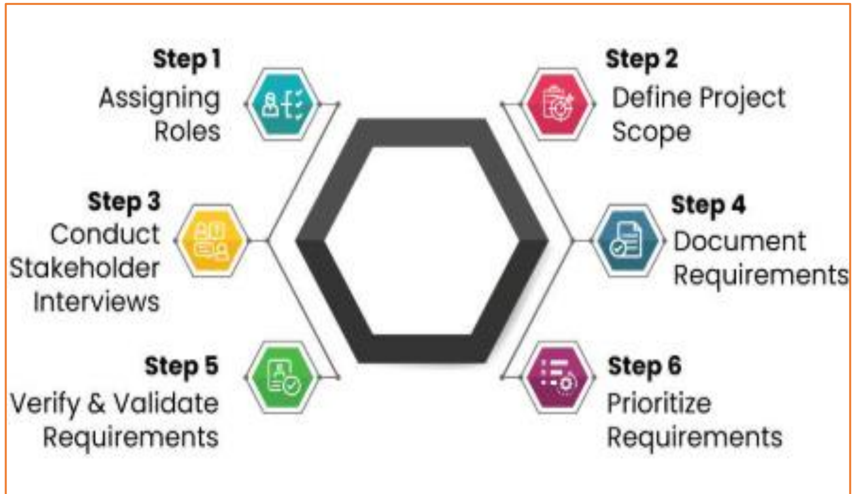
- Record requirements as user stories, use cases, or specifications.
- Include functional and non-functional requirements in documentation.

Step 5: Verify and Validate Requirements

- Ensure requirements align with stakeholder intentions.
- Validate that requirements meet project goals and objectives.

Step 6: Prioritize Requirements

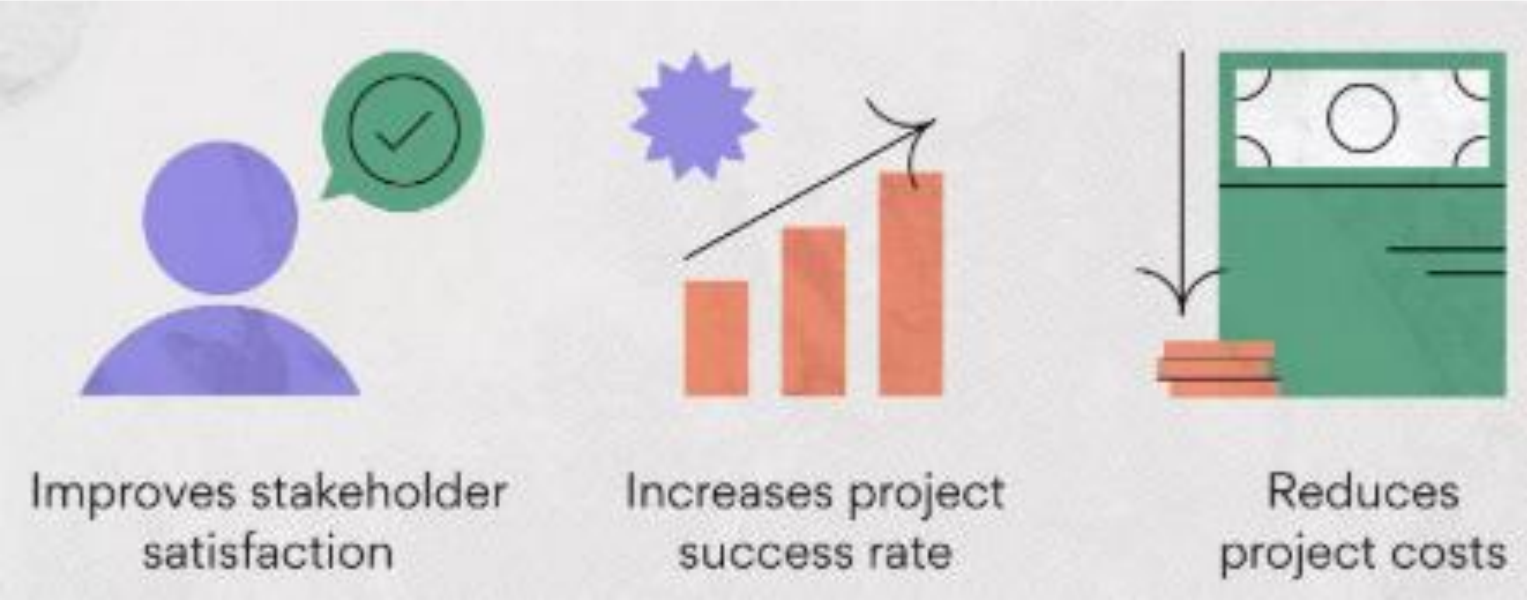
- **Rank requirements** by importance and project constraints.
- Create a **development roadmap** focusing on critical features.



Contd.

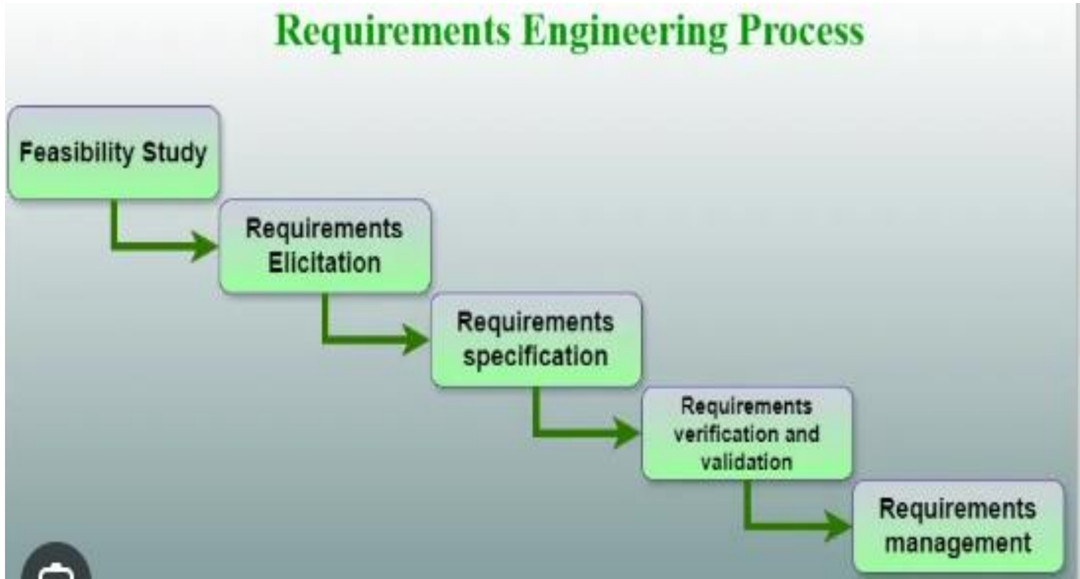
Requirement Gathering Benefits

- Improved Communication
- Efficient Resource Utilization
- Enhanced Quality
- Risk Management
- Accurate Planning


Improves stakeholder satisfaction
Increases project success rate
Reduces project costs

Requirement Analysis and specification: Overview

- **Goal:** Clearly define and document all customer requirements.
- Begins after completing the feasibility study phase.
- Ensures project is financially and technically feasible.
- Ends with a reviewed Software Requirements Specification (SRS) document.
- Users often have incomplete system views
- Organizes customer requirements systematically into the SRS document.
- Engineers performing analysis are called **Analysts**


Requirements Engineering Process
Feasibility Study
Requirements Elicitation
Requirements specification
Requirements verification and validation
Requirements management

Contd.

Who Performs Requirements Analysis?

- Conducted ***by experienced development team members***.
- Engineers spend time at the customer site.
- ***System analysts*** handle requirements gathering and documentation.

Two main activities:

- Requirements gathering and analysis.
- Requirements specification.

- **Requirements Analysis -** Problem Identification
- **<u>Purpose</u> of Requirements Analysis**

Clarify customer requirements and resolve gathered problems.

Contd.

Problems of Requirement Analysis

- **Unclear Stakeholder Needs:**

  Stakeholders often struggle to define their actual requirements.

- **Vague Communication:**

  Requirements are expressed using ambiguous, domain-specific language.

- **Conflicting Requirements:**

  Different stakeholders may provide contradictory expectations.

- **Organizational Influence:**

  Political and organizational factors can impact requirements.

- **Changing Requirements:**

  Requirements tend to evolve during the analysis phase.

Requirement Specification

- **SRS Document**

  Requirement collected gets transformed into structure SRS document.

- **Define Requirements**

  It defines the requirements of the proposed system.

- **Analysis to Documentation**

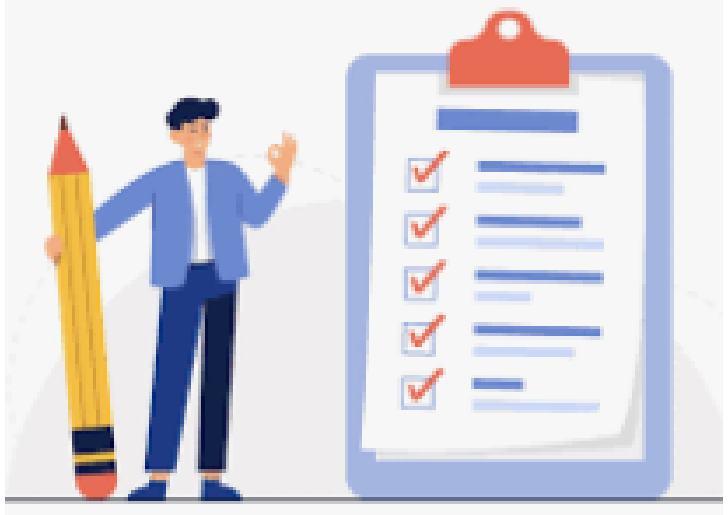
  Analyzed requirement must be clearly documented.

- **Clarity for Development**

  Essential to clarify project development needs.

Customer

Service Provider



Basic

- SRS is a description of a s/w system to be developed.

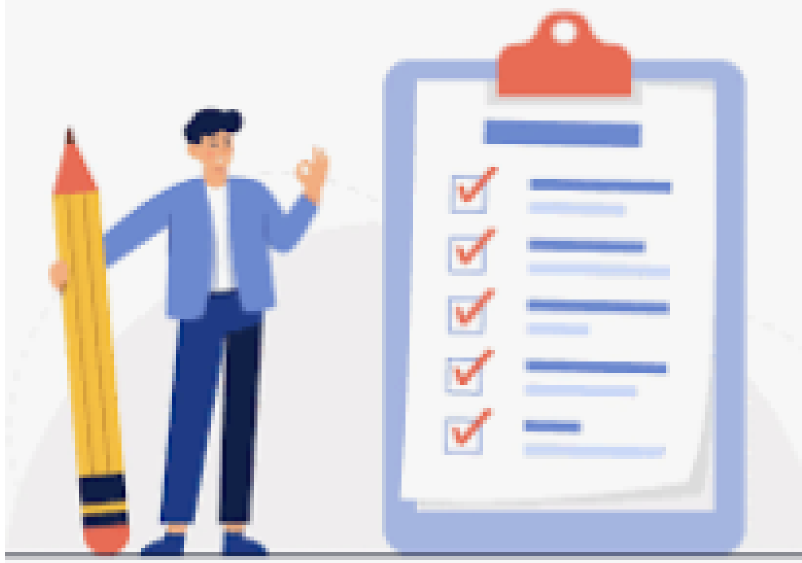
E.g. UMS.

- It lays out functional and non-functional requirements of the s/w to be developed.
- It may ***include a set of <u>use cases</u>*** (situation e.g. student forgot the password so how to recover i.e. **forgot password**; student wants to discuss doubts with faculty so through which discussion framework)that describe user interaction that the software must provide to the user for perfect interaction.

Contd.

Software Requirements Specification (SRS)

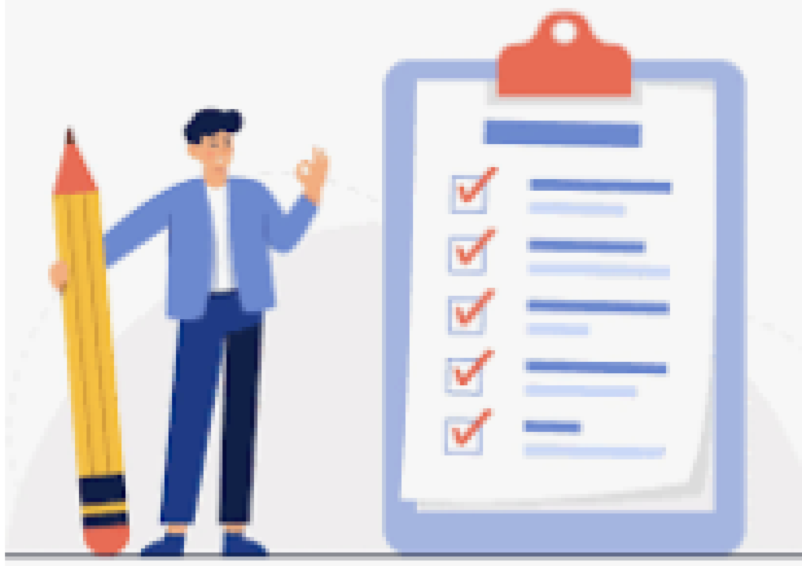
- Organizes requirements into a structured SRS document
- Serves as a contract between customer and team
- Ensures product meets documented requirements
- Acts as a reference for development
- Specifies external behavior of the system
- Written using end-user terminology
- SRS is designed before going to the design phase.



Contd.

Properties of a Good SRS

- Concise and free from ambiguity
- Focuses on "what," not "how"
- Easy to modify and well-structured
- Must be consistent and complete
- Should be traceable and verifiable



Contd.

SRS Document Parts

- **Functional Requirements**:

  Describes system functions and data transformation.

- **Non-Functional Requirements**:

  Reliability, performance, and interface constraints.

- **System Constraints**:

  Standards, hardware, OS, and data compliance.

IEEE format for SRS Document

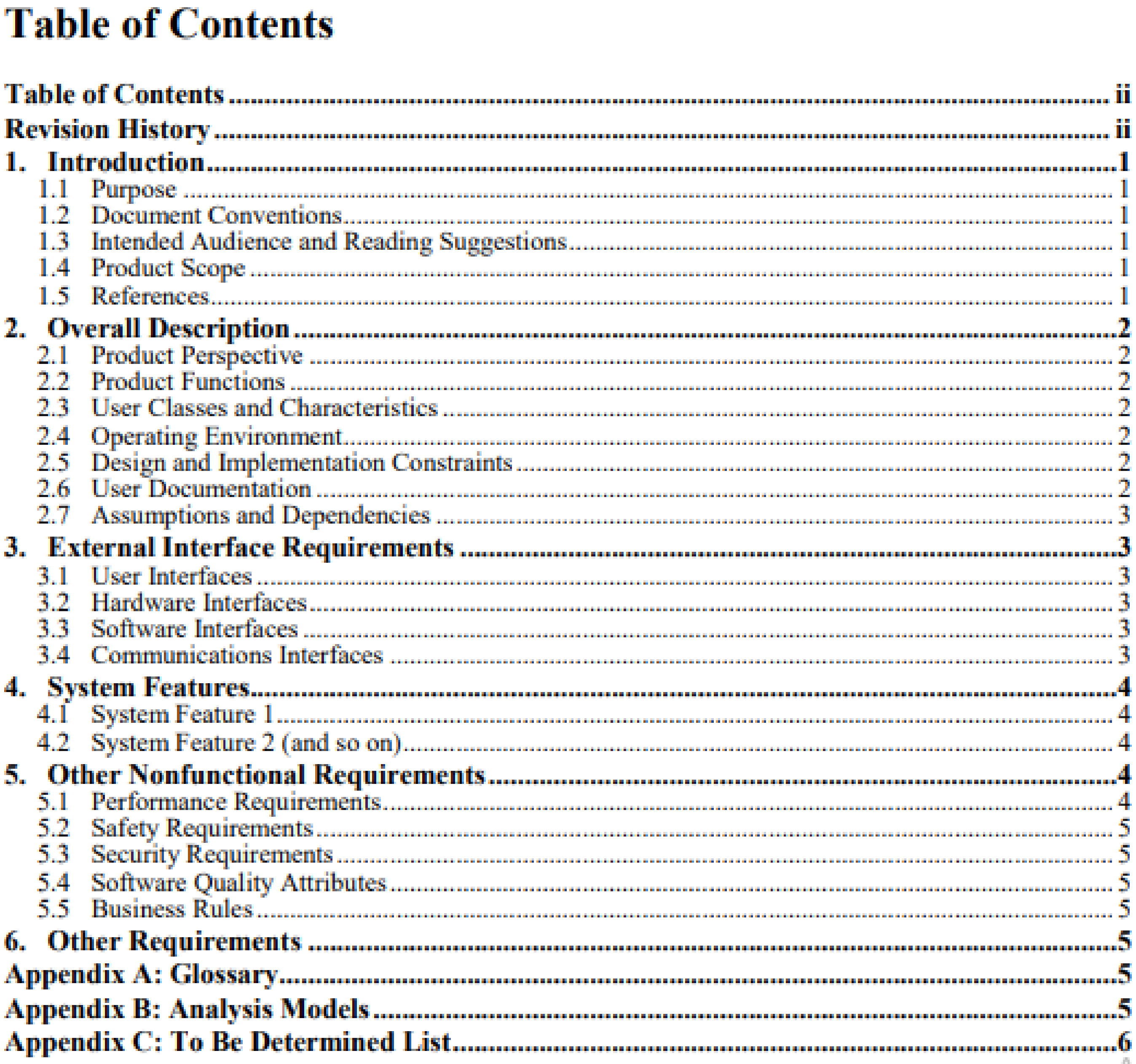


Table of Contents

1. What is the primary purpose of a Feasibility Study in software engineering?

A) To design the system architecture
B) To assess whether the project is technically and financially viable
C) To gather requirements from stakeholders
D) To test the system’s functionalities

2. Which of the following is NOT a type of feasibility considered in a Feasibility Study?

A) Technical Feasibility
B) Design Feasibility
C) Operational Feasibility
D) Economic Feasibility

3. In the context of Requirement Gathering, who are the primary stakeholders typically involved in the process?

A) Developers only
B) Project managers and designers
C) Clients, end-users, and business analysts
D) Software testers and quality assurance engineers

4. Which of the following techniques is commonly used for gathering requirements from stakeholders?

A) Prototyping
B) Waterfall model
C) Brainstorming
D) Code reviews

5. In which phase of software engineering is Requirement Analysis typically performed?

A) Implementation
B) Design
C) Maintenance
D) Requirements gathering and analysis

6. What is the main focus of Requirement Analysis in software engineering?

A) To identify and document the requirements for the software system
B) To test and validate the software
C) To design the system architecture
D) To implement the software functionality

7. Which document typically specifies the detailed software requirements and functional specifications?

A) System design document
B) Software Requirement Specification
C) User manual
D) Test case document

8. What is the main goal of creating a Software Requirement Specification (SRS) document?

A) To define the system's architecture
B) To give a comprehensive understanding of system requirements and guide future stages of development
C) To list the tools and technologies used in development
D) To provide test cases for software testing

- **Which type of requirement focuses on how the system will perform under specific conditions?**

A) Functional requirements
B) Non-functional requirements
C) Technical requirements
D) Business requirements

- **During the analysis phase, what is typically created to represent system interactions?**

A) User interface
B) Data models
C) Use case diagrams and flowcharts
D) Source code